IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant:	David E. LOWELL	§	Confirmation No.:	8180
Serial No.:	10/677,159	§	Group Art Unit:	2195
Filed:	10/01/2003	§	Examiner:	Meng Yao Zhe
For:	Runtime Virtualization And Devirtualization Of Memory By A Virtual Machine Monitor	§	Docket No.:	200300561-1

APPEAL BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Date: February 5, 2009

Sir:

Appellant hereby submits this Appeal Brief in connection with the above-identified application. A Notice of Appeal was electronically filed on December 5, 2008.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, L.P. (HPDC), a Texas Limited Partnership, having its principal place of business in Houston, Texas. HPDC is a wholly owned affiliate of Hewlett-Packard Company (HPC). The Assignment from the inventor to HPDC was recorded on December 2, 2003, at Reel/Frame 014168/0718.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals or interferences.

III. STATUS OF THE CLAIMS

Originally filed claims:	1-67.
Claim cancellations:	None.
Added claims:	None.
Presently pending claims:	1-67.
Presently appealed claims:	1-67.

IV. STATUS OF THE AMENDMENTS

No claims were amended after the final Office Action dated October 6, 2008.

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

This section provides a concise explanation of the subject matter defined in each of the independent claims, referring to the specification by page and line number or to the drawings by reference characters as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified with a corresponding reference to the specification or drawings where applicable. The specification references are made to the application as filed by Applicants. Note that the citation to passages in the specification or drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element. Also note that these specific references are not exclusive; there may be additional support for the subject matter elsewhere in the specification and drawings.

In accordance with the invention of claim 1, a method[1] of running a virtual machine monitor[2] on computer hardware,[3] the hardware including memory,[4] the method comprises waiting to commence virtualization[5] of the memory until runtime.[6]

In accordance with the invention of claim 13, a method[7] of running a virtual machine monitor[8] on computer hardware[9] and an operating system[10] on the virtual machine monitor, the hardware including memory,[11] the memory

---

[1] Fig. 2. Disclosure p. 6 line 1 of para. [0026].

[2] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[3] Fig. 1, hardware 110. Disclosure p. 5 line 2 of para. [0023].

[4] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[5] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030].

[6] Disclosure p. 5 lines 1-4 of para. [0025].

[7] Fig. 2. Disclosure p. 6 line 1 of para. [0026].

[8] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[9] Fig. 1, hardware 110. Disclosure p. 5 line 2 of para. [0023].

[10] Fig. 1, OS 112. Disclosure p. 5 line 1 of para. [0024].

[11] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

virtualized by the virtual machine monitor,[12] the method comprises devirtualizing the memory at runtime.[13]

In accordance with the invention of claim 28, a computer[14] comprises memory[15] including first and second portions. The first portion is encoded with a virtual machine monitor[16] that waits to commence virtualization of the second portion until runtime.[17]

In accordance with the invention of claim 36, an article for a computer comprises computer memory[18] including a first portion encoded with a virtual machine monitor[19] that waits to commence virtualization of a second portion of the memory until runtime.[20]

In accordance with the invention of claim 44, a computer[21] comprises hardware[22] including memory[23] and a virtual machine monitor[24] for virtualizing the memory and devirtualizing the memory at runtime.[25] The virtual machine monitor virtualizes the memory when multiple operating system instances are running and devirtualizes the memory when a single operating system instance is running.[26]

---

[12] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030].

[13] Fig. 2, 220. Disclosure p. 7 line 6 of para. [0030].

[14] Fig. 1, computer 100. Disclosure p. 5 line 1 of para. [0023].

[15] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[16] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[17] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030]; p. 8 line 1 of para. [0032] through p. 12 line 4 of para. [0046].

[18] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[19] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[20] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030]; p. 8 line 1 of para. [0032] through p. 12 line 4 of para. [0046].

[21] Fig. 1, computer 100. Disclosure p. 5 line 1 of para. [0023].

[22] Fig. 1, hardware 110. Disclosure p. 5 line 2 of para. [0023].

[23] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[24] Fig. 1, VMM 114. Disclosure p. 5 line 2 of para. [0024].

[25] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030]; p. 8 line 1 of para. [0032] through p. 12 line 4 of para. [0046].

[26] Fig. 2, 218. Disclosure p. 7 line 5 of para. [0030]; p. 8 line 1 of para. [0032] through p. 12 line 4 of para. [0046].

In accordance with the invention of claim 56, an article for a computer[27] including hardware,[28] the hardware including computer memory,[29] the article comprises memory encoded with software for devirtualizing the computer memory at runtime.[30]

[27] Fig. 1, computer 100. Disclosure p. 5 line 1 of para. [0023].

[28] Fig. 1, hardware 110. Disclosure p. 5 line 2 of para. [0023].

[29] Fig. 1, memory. Disclosure p. 5 line 4 of para. [0023].

[30] Fig. 2, 220. Disclosure p. 7 line 6 of para. [0030].

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-43 are directed to non-statutory subject matter under 35 U.S.C. § 101.

Whether claims 1-43 are indefinite under 35 U.S.C. § 112, second paragraph.

Whether claims 1, 2, 4, 5, 10, 12-19, 25, 27-30, 35-38, 43-50, 55-62 and 67 are anticipated by Bean (U.S. Pat. No. 4,843,541).

Whether claims 6-9, 11, 26, 31-34, 39-42, 54 and 66 are obvious over Bean under 35 U.S.C. § 103.

Whether claims 20-22, 51-53 and 63-65 are obvious over Bean in view of Chu (U.S. Pat. No. 6,256,657).

Whether claims 23 and 24 are obvious over Bean in view of Waldspurger (U.S. Pat. No. 6,789,156).

Whether claim 3 is obvious over Bean in view of Bugnion (U.S. Pat. No. 6,296,847).

VII. ARGUMENT

A. Rejection of claims 1-43 under § 101

Claim 1 is directed to a method of running a virtual machine on "computer hardware, the hardware including memory… ." The CAFC held that a method claim that is tied to a particular machine is statutory under § 101. In re Bilski, 545 F.3d 943, 961 (Fed. Cir. 2008). The CCPA, whose opinions like those of the CAFC are binding on the Patent Office, held that a claimed method that was performed by a digital computer was statutory. Application of Bernhart, 417 F.2d 1395, 1398-1401 (CCPA 1969) (holding that the apparatus and method claims were directed to statutory subject matter). The Bernhart decision also states that a digital computer, programmed to perform a novel and unobvious function is, in fact, a new machine, or at least an improvement to an old machine, and thus statutory. Id. at 1400. Specifically, "if a machine is programmed in a certain new and unobvious way, it is physically different from the machine without that program; its memory elements are differently arranged." Id. at 1400. The Federal Circuit referred to Bernhart favorably and held that a general purpose computer, or microprocessor, programmed to carry out an algorithm becomes a special purpose computer and thus a new machine. WMS Gaming, Inc. v. International Game Technology, 184 F.3d 1339, 1348 (Fed. Cir. 1999) ("A general purpose computer, or microprocessor, programmed to carry out an algorithm creates "a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software."). Accordingly, under the binding precedent of Bernhart and WMS Gaming, a computer that is programmed to perform a novel and unobvious method is, by definition, a particular machine for purpose of the § 101 analysis under Bilski. See also Ex Parte Wang, 2008 WL 4448241,*7, n.3 (Bd.Pat.App. & Interf. Sept. 30, 2008) (citing approvingly Application of Bernhart).

Claim 1 requires "computer hardware" and "memory." Such components comprise a particular machine per the Bilski and Bernhart decisions. Additionally, claim 1 does not merely recite steps that can be performed by a human, a factor

that militated in favor of a finding in Bernhart that the method claim was statutory. For at least these reasons, claim 1 and its dependent claims are directed to statutory subject matter.

The same or similar analysis applies to claim 13 and its dependent claims.

Claim 28 is directed to a "computer comprising memory." Application of Bernhart held that a digital computer, programmed to perform in a novel and unobvious way, is a new machine or at least a new and useful improvement of an unprogrammed machine. Application of Bernhart, 417 F.2d at 1400. Based on Bernhart, claim 28, which also is directed to a computer programmed to perform in a novel and unobvious way, is directed to statutory subject matter. See id. at 1400 ("[programmed] machines are statutory under 35 U.S.C. § 101").

The same or similar analysis applies to claim 36 and its dependent claims.

B. Rejection of claims 1-43 under § 112, second paragraph

The claims in this group do not all stand and fall together. The Examiner rejected claims 1-43, which refer to "runtime." The Examiner expressed some uncertainty as to what runtime refers. Appellant's disclosure clearly defines "runtime" as follows: "Runtime is the period of normal execution of the operating system after boot and before shutdown." Appellant has been its own lexicographer and one of ordinary skill in the art would clearly understand the scope of the claims in terms of the word "runtime."

The Examiner also expressed some uncertainty as to the phrase "waiting to commence virtualization until runtime." The Examiner asked if the system is "just stalling before runtime?" Final Office Action page 3. The claims require waiting for runtime. Appellant respectfully submits that that phrase is very clear and certainly sufficiently clear to one of ordinary skill in the art. For example, in claim 1, virtualization does not begin until runtime. That is, virtualization does not occur during the boot process before the operating system is loaded.

C. Anticipation rejection of clams 1, 2, 4, 5, 10, 12-19, 25, 27-30, 35-38, 43-50, 55-62 and 67 over Bean

Claim 1 requires "waiting to commence virtualization of the memory until runtime." Appellant explained above what is meant by "runtime" which is the

period of normal execution of the operating system after boot and before shutdown. Bean describes dynamically reassigning resources to different operating systems (see e.g., col. 22, lines 50-60), but does not indicate that virtualization does not commence until runtime as in claim 1. For at least this reason, the Examiner erred in rejecting claim 1 and its dependent claims over Bean.

Claim 13, in part, requires "devirtualizing the memory at runtime." The Examiner cites Bean at col. 22, lines 47-55 and col. 29, lines 33-55 as teaching this limitation. See Office Action dated 04/08/08, page 5, item 12. Bean appears to describe dynamic reassignment of resources without devirtualizing the memory. Switching a partition from one guest to another as in Bean involves reassigning virtual memory rather than devirtualizing memory as in claim 13. One of ordinary skill in the art would understand that Bean's reassignment of virtual memory does not reduce virtualization overhead as does Applicant's devirtualization. For at least these reasons, the Examiner erred in rejecting claim 13 and its dependent claims over Bean.

Amended claim 28, in part, requires "memory including first and second portions, the first portion encoded with a virtual machine monitor that waits to commence virtualization of the second portion until runtime." For much the same reason as given for claim 1, Bean does not teach the above limitations. For at least this reason, the Examiner erred in rejecting claim 28 and its dependent claims.

Amended claim 36, in part, requires "computer memory including a first portion encoded with a virtual machine monitor that waits to commence virtualization of a second portion of the memory until runtime." For much the same reason as given for claim 1, Bean does not teach the above limitations. For at least this reason, the Examiner erred in rejecting claim 36 and its dependent claims.

Amended claim 44 requires "a virtual machine monitor for virtualizing the memory and devirtualizing the memory at runtime, wherein the virtual machine monitor virtualizes the memory when multiple operating system instances are

running and devirtualizes the memory when a single operating system instance is running." For much the same reasons as given for claim 13, Bean does not teach devirtualizing memory at runtime as in claim 44. Further, Bean does not teach devirtualizing memory when a single operating system instance is running as in claim 44. Instead, Bean appears to assume that multiple operating systems will be running (see e.g., col. 7, lines 57-61). For at least these reasons, the Examiner erred in rejecting claim 44 and its dependent claims.

Claim 56, in part, requires "memory encoded with software for devirtualizing the computer memory at runtime." For much the same reasons as given for claim 13, Bean does not teach devirtualizing the computer memory at runtime as in claim 56. For at least this reason, the Examiner erred in rejecting claim 56 and its dependent claims.

D. Obviousness rejection of clams 6-9, 11, 26, 31-34, 39-42, 54 and 66 over Bean

Claims 6-9, 11, 26, 31-34, 39-42, 54 and 66 depend from independent claims which, as explained above, are patentable over Bean. Bean does not teach or suggest the limitations discussed above in the preceding section. For at least that reason, the Examiner erred in rejecting claims 6-9, 11, 26, 31-34, 39-42, 54 and 66.

E. Obviousness rejection of clams 20-22, 51-53 and 63-65 over Bean in view of Chu

Claims 20-22, 51-53 and 63-65 depend from independent claims which, as explained above, are patentable over Bean. Chu does not satisfy the deficiencies of Bean. For at least that reason, the Examiner erred in rejecting claims 20-22, 51-53 and 63-65.

F. Obviousness rejection of clams 23 and 24 over Bean in view of Waldspurger

Claims 23 and 24 depend from independent claim 13 which, as explained above, is patentable over Bean. Waldspurger does not satisfy the deficiencies of Bean. For at least that reason, the Examiner erred in rejecting claims 23 and 24.

G. Obviousness rejection of clam 3 over Bean in view of Chu

Claim 3 depends from allowable claim 1. Chu does not satisfy the deficiencies of Bean. For at least that reason, the Examiner erred in rejecting claim 3.

H. Conclusion

For the reasons stated above, Appellant respectfully submits that the Examiner erred in rejecting all pending claims. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,

/Jonathan M. Harris/

Jonathan M. Harris
PTO Reg. No. 44,144
CONLEY ROSE, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
ATTORNEY FOR APPELLANT

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400

VIII. CLAIMS APPENDIX

1. (Previously presented) A method of running a virtual machine monitor on computer hardware, the hardware including memory, the method comprising waiting to commence virtualization of the memory until runtime.

2. (Original) The method of claim 1, wherein the virtualization includes constructing an Identity mapping of physical to machine memory; and commencing to use the virtual machine monitor at runtime to manage memory translation.

3. (Original) The method of claim 2, wherein the Identity mapping is constructed prior to runtime.

4. (Original) The method of claim 2, wherein the memory translation is initially performed according to the Identity mapping.

5. (Original) The method of claim 4, wherein the virtual machine monitor modifies the mapping after the physical memory has been virtualized.

6. (Original) The method of claim 2, wherein an operating system is running on the virtual machine monitor prior to virtualizing the memory; and wherein the memory translation is managed by allowing the operating system to define virtual-to-physical mapping, and the virtual machine monitor to define physical-to machine mapping.

7. (Original) The method of claim 6, wherein the virtual machine monitor dynamically composes virtual-to-physical translations with physical-to-machine translations.

8. (Original) The method of claim 6, wherein the virtual machine monitor inspects the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

9. (Original) The method of claim 6, further comprising loading a translation lookaside buffer with virtual-to-machine translations.

10. (Previously presented) The method of claim 1, wherein only a portion of the memory is virtualized at runtime.

11. (Original) The method of claim 1, wherein the hardware includes a CPU that was virtualized prior to the virtualization of the memory.

12. (Original) The method of claim 1, further comprising performing runtime devirtualization of the virtualized memory.

13. (Original) A method of running a virtual machine monitor on computer hardware and an operating system on the virtual machine monitor, the hardware including memory, the memory virtualized by the virtual machine monitor, the method comprising devirtualizing the memory at runtime.

14. (Original) The method of claim 13, wherein a portion of the memory is devirtualized.

15. (Original) The method of claim 13, wherein when the operating system is booted, the virtual machine monitor exposes the booting operating system to physical memory no larger than machine memory, where the physical memory does not span any memory holes.

16. (Previously presented) The method of claim 13, wherein the operating system defines virtual-to-physical translations prior to the runtime devirtualization;

wherein the virtual machine monitor defines physical-to-machine translations prior to the runtime devirtualization; wherein the virtual machine monitor composes dynamically the virtual-to-physical translations with the physical-to-machine translations prior to the runtime devirtualization, wherein the runtime devirtualization includes having the virtual machine monitor cease to perform the dynamic composition of translations.

17. (Original) The method of claim 13, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using the operating system to manage address translation with respect to the devirtualized memory.

18. (Original) The method of claim 17, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

19. (Original) The method of claim 17, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

20. (Original) The method of claim 17, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

21. (Original) The method of claim 20, further comprising preventing the physical-to-machine mapping from being modified during the remapping, and temporarily preventing some or all write accesses to memory.

22. (Original) The method of claim 17, wherein the operating system and any application activity is paused while the remapping is performed.

23. (Original) The method of claim 17, further comprising maintaining a back map that contains for each page of machine memory a list of the pages of physical memory that map to it, and a list of free machine pages.

24. (Original) The method of claim 17, wherein the remapping is performed without a back map by maintaining a reference count for each machine page is kept, and freeing machine pages when their reference counts are zero.

25. (Original) The method of claim 17, wherein the remapping is performed without a back map by constructing a list of the physical pages mapping to a page of machine memory by searching the physical-to-machine mapping.

26. (Original) The method of claim 17, wherein managing the address translation includes having the virtual machine monitor cease to inspect the operating system's virtual-to-physical translations; and ceasing to maintain a page table of direct virtual-to-machine mappings.

27. (Original) The method of claim 17, wherein managing the address translation includes having the virtual machine monitor cease to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.

28. (Previously presented) A computer comprising memory including first and second portions, the first portion encoded with a virtual machine monitor that waits to commence virtualization of the second portion until runtime.

29. (Original) The computer of claim 28, wherein the virtualization includes constructing an Identity mapping of physical to machine memory; and commencing to use the virtual machine monitor at runtime to manage memory translation.

30. (Original) The computer of claim 29, wherein the virtual machine monitor modifies the mapping after the physical memory has been virtualized.

31. (Original) The computer of claim 29, wherein an operating system is running on the virtual machine monitor prior to virtualizing the memory; and wherein the memory translation is managed by allowing the operating system to manage virtual-to-physical mapping, and allowing the virtual machine monitor to manage physical-to machine mapping.

32. (Original) The computer of claim 31, wherein the virtual machine monitor dynamically composes virtual-to-physical translations with the physical-to-machine translations.

33. (Original) The computer of claim 31, wherein the virtual machine monitor inspects the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

34. (Original) The computer of claim 31, wherein a translation lookaside buffer is loaded with the virtual-to-machine translations.

35. (Original) The computer of claim 28, wherein only a portion of physical memory is virtualized at runtime.

36. (Previously presented) An article for a computer, the article comprising computer memory including a first portion encoded with a virtual machine monitor that waits to commence virtualization of a second portion of the memory until runtime.

37. (Original) The article of claim 36, wherein the virtualization includes constructing an Identity mapping of physical to machine memory; and

commencing to use the virtual machine monitor at runtime to manage memory translation.

38. (Original) The article of claim 37, wherein the virtual machine monitor can modify the mapping after the physical memory has been virtualized.

39. (Original) The article of claim 37, wherein the memory translation is managed by allowing an operating system to manage virtual-to-physical mapping, and allowing the virtual machine monitor to manage physical-to machine mapping.

40. (Original) The article of claim 39, wherein the virtual machine monitor can dynamically compose virtual-to-physical translations with the physical-to-machine translations.

41. (Original) The article of claim 39, wherein the virtual machine monitor can inspect the virtual-to-physical mappings by the operating system and maintains page tables of virtual-to-machine mappings.

42. (Original) The article of claim 37, wherein the virtual machine monitor can load a translation lookaside buffer with virtual-to-machine translations.

43. (Original) The article of claim 36, wherein the virtual machine monitor can virtualize only a portion of physical memory at runtime.

44. (Previously presented) A computer comprising hardware including memory; and a virtual machine monitor for virtualizing the memory and devirtualizing the memory at runtime, wherein the virtual machine monitor virtualizes the memory when multiple operating system instances are running and devirtualizes the memory when a single operating system instance is running.

45. (Original) The computer of claim 44, wherein a portion of the memory is devirtualized.

46. (Original) The computer of claim 44, wherein when an operating system is booted, the virtual machine monitor exposes the booting operating system to physical memory no larger than machine memory, where the physical memory does not span any memory holes.

47. (Previously presented) The computer of claim 44, wherein an operating system defines virtual-to-physical translations prior to the runtime devirtualization; wherein the virtual machine monitor defines physical-to-machine translations prior to the runtime devirtualization; wherein the virtual machine monitor composes dynamically the virtual-to-physical translations with the physical-to-machine translations prior to the runtime devirtualization; wherein the runtime devirtualization includes having the virtual machine monitor cease to perform the dynamic composition of translations.

48. (Original) The computer of claim 44, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using an operating system to manage address translation with respect to the devirtualized memory.

49. (Original) The computer of claim 48, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

50. (Original) The computer of claim 48, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

51. (Original) The computer of claim 48, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

52. (Original) The computer of claim 51, wherein the physical-to-machine mapping is prevented from being modified during the remapping, and some or all write accesses to memory are temporarily prevented.

53. (Original) The computer of claim 48, wherein the operating system and any application activity is paused while the remapping is performed.

54. (Original) The computer of claim 48, wherein managing the address translation includes having the virtual machine monitor cease to inspect the operating system's virtual-to-physical translations; and wherein maintenance of a page table of direct virtual-to-machine mappings is ceased.

55. (Original) The computer of claim 48, wherein managing the address translation includes having the virtual machine monitor cease to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.

56. (Original) An article for a computer including hardware, the hardware including computer memory, the article comprising memory encoded with software for devirtualizing the computer memory at runtime.

57. (Original) The article of claim 56, wherein the software causes a portion of the memory to be devirtualized.

58. (Original) The article of claim 56, wherein the software includes a virtual machine monitor; and wherein when an operating system is booted on the virtual

machine monitor, the virtual machine monitor exposes the booting operating system to physical memory no larger than machine memory, where the physical memory does not span any memory holes.

59. (Original) The article of claim 56, wherein an operating system defines virtual-to-physical translations prior to the runtime devirtualization; wherein the software includes a virtual machine monitor for defining physical-to-machine translations prior to the runtime devirtualization, composing dynamically the virtual-to-physical translations with the physical-to-machine translations prior to the runtime devirtualization, and ceasing to perform the dynamic composition of translations during the runtime virtualization; and wherein after the runtime devirtualization is performed, memory translation is performed by directly using the virtual-to-physical mapping defined by the operating system.

60. (Original) The article of claim 56, wherein the devirtualization includes remapping physical memory so a physical-to-machine mapping becomes an Identity mapping; and using an operating system to manage address translation with respect to the devirtualized memory.

61. (Original) The article of claim 60, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped directly.

62. (Original) The article of claim 60, wherein pages of physical memory that are already Identity-mapped are not remapped, and wherein at least some other pages of physical memory are remapped indirectly.

63. (Original) The article of claim 60, wherein the remapping of the physical memory is performed concurrently with operating system and application activity.

64. (Original) The article of claim 63, wherein the physical-to-machine mapping is prevented from being modified during the remapping, and some or all write accesses to memory are temporarily prevented.

65. (Original) The article of claim 60, wherein the operating system and any application activity is paused while the remapping is performed.

66. (Original) The article of claim 60, wherein the software includes a virtual machine monitor that manages the address translation by ceasing to inspect the operating system's virtual-to-physical translations; and wherein maintenance of a page table of direct virtual-to-machine mappings is ceased.

67. (Original) The article of claim 60, wherein the software includes a virtual machine monitor for managing the address translation by ceasing to compose dynamically the operating system's virtual-to-physical translations with the virtual machine monitor's physical-to-machine translations for a portion of physical memory that is devirtualized.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.